| Document Title | Assignment 01 |
|---|---|
| Course | CS 200 Introduction to Programming |
| Academic Year | 2016-2017 |
| Semester | Fall |
| Due Date | September 27, 2016 11:55 pm |
| Marks | 100 |

The assignment is due on September 27, 2016 11:55 pm. The late submission policy of 15% deduction per day up to 2 days applies. This assignment will require reasonable amount of time so try to start the assignment as early as you can.

Please keep in mind the following guidelines:

- Do not share your program code with anyone.
- Do not copy code from the internet.
- If you receive any assistance, mention the part of code in which you received assistance.
- You must be able to explain any part of your submitted code.
- All submissions are subject to automated plagiarism detection.

## What to submit:

You have to submit .cpp files containing source code.  Zip all .cpp files into one file named as <your 8 digit roll number>.zip and submit the zip file.

# Task 1 (20 marks)

Write a program to manage operations at PDC. Your program has 5 pre defined food items of your own choice. Each food item has a standard, half and quarter portion and each portion has its own price. User can select at most three items.

The flow of program is as following:
Program shows the list of available food items.
User selects desired food item.
User selects desired portion of food item.
Program should keep running until **sentinel value** is entered or item count becomes three.

Output:

```
            Date: September 19, 2016
            Time: 12:00 pm
       ***********************
Sr. No       Item        Portion       Price
1            Chicken      Half          100
2            Rice         Full          50
3            Salad        Quarter       25


            Item Count:              3
            TOTAL:                   175
```

The output should be left aligned and should contain current date and time as shown above.

# Task 2 (30 marks)

Write a program that performs calculation for fractional numbers. A fractional number has a numerator part and a denominator part and written as **numerator/denominator**. You have to take numerator and denominator of fractional numbers as integer input separately, perform the desired operation and output the result in fractional form. Supported operations are division, multiplication, addition, subtraction and inverse (in case of inverse operation, only one fractional number is required).

The flow of program is as following:
Program shows the list of supported operations .
User selects desired operation and enters operand(s).
Program shows the result for the corresponding operation.
Program should keep running until **sentinel value** is entered.

Sample Run:

Press 1 for addition
Press 2 for subtraction
Press 3 for multiplication
Press 4 for division
Press 5 for inverse
Press -1 to exit
Enter your choice: 1
Enter numerator for first fractional number: 15
Enter denominator for first fractional number: 0

Error…!! Denominator can't be zero..!!

Enter denominator for first fractional number: 25
Enter numerator for second fractional number: 20
Enter denominator for second fractional number: 25

    **** RESULT ****
    15/25 + 20/25 = 7/5
    **************


Result of calculation should be in simplest form i-e (15/25 + 20/25 = 35/25 = 7/5).The program should keep running until -1 is entered. **Do handle division by zero case.**

# Task 3 (50 marks)

Write a program that performs number system conversion. You have to deal with binary, decimal and hexadecimal number systems. Program takes input in any of above mentioned number system as desired by user and outputs equivalent value in other number systems. **You need to study string traversal in detail to do this task (Read Section 2.5 from book 'C++ for everyone'; http://www.cplusplus.com/reference/string/string/operator[]/).** (Do not use built in function of c++ to perform conversion).

The flow of program is as following:
Program shows the list of supported number systems.
User selects desired number system and enters input.
Program shows equivalent values in other number systems corresponding to the given input.
Program should keep running until **sentinel value** is entered.

Sample Run:

Enter 1 for decimal number system
Enter 2 for binary number system
Enter 3 for hexadecimal number system
Enter  -1 to exit
Enter your choice : 3
Enter a hexadecimal number : f0a87

Decimal : 985735
Binary : 11110000101010000111
Hexadecimal : f0a87

Assume that input number can be of 5 digits at maximum. In case of hexadecimal numbers, input is not case sensitive. For example f0a87, F0a87 and f0A87 should be treated as same number. Moreover, you have to show proper error messages if invalid input is entered.